Investigating Allocation of Heterogeneous Storage Resources on HPC Systems

Julien Monniot, François Tessier, Gabriel Antoniu - Team KerData@INRIA - France

7th Per3s Workshop - 30/05/2023

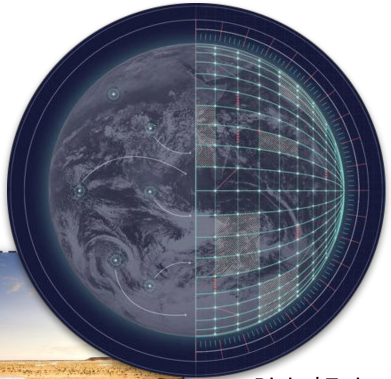
01

Context and motivations

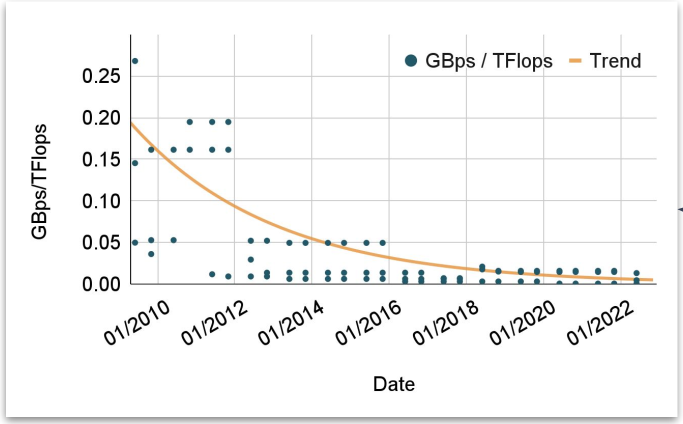
Growing I/O requirements

Data deluge from new large-scale scientific workflows
- ➔ Multiplication of data sources
- ➔ Drastic increase from scientific projects

SKA Radio Telescope - skatelescope.org



Digital Twin
ecmwf.int



↗ PFlops ↘ TBps

=

↗ gap between compute and I/O performances

Compute-centric to **data-centric** shift

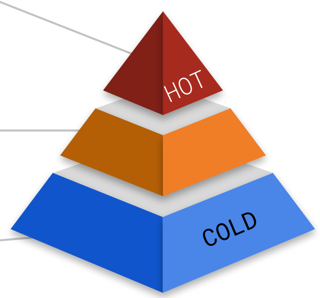
↗ **I/O pressure** for large-scale systems

Current trends

Node-local / Platform integrated (SSD, NVRAM, …)

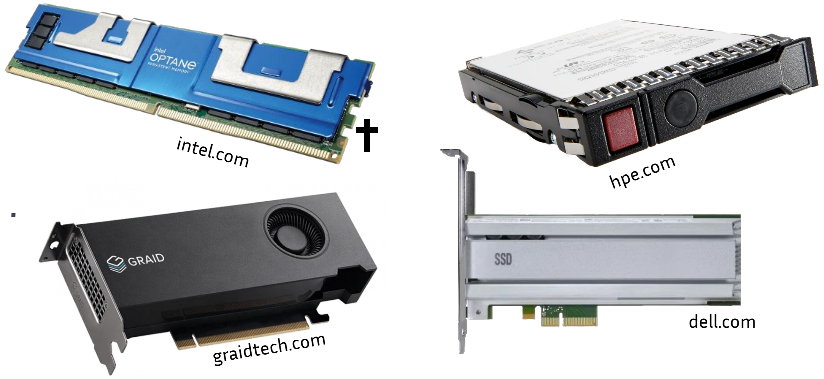
Burst-buffers, scratch/staging area (SSD, NVMeoF, HDD, …)

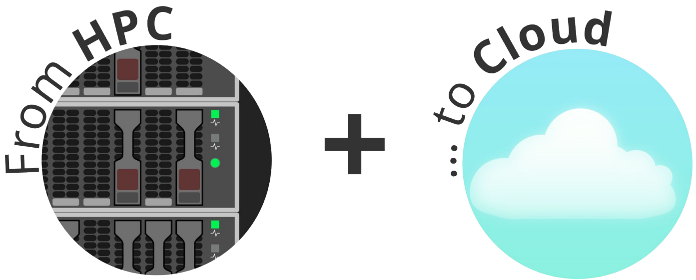
PFS/Archives (HDD, tapes)



- Deep storage **hierarchy**

- **New** underlying **storage technologies**

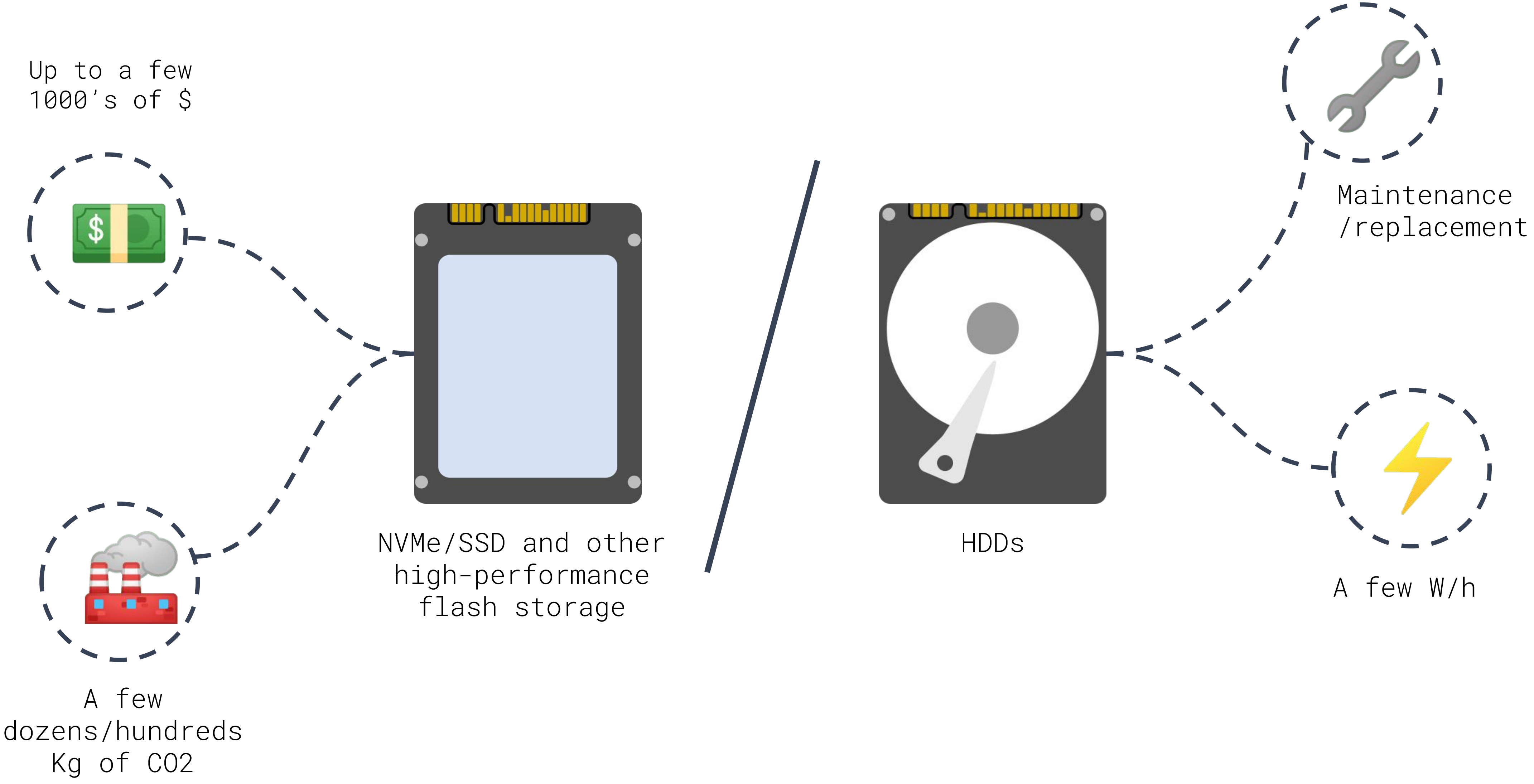




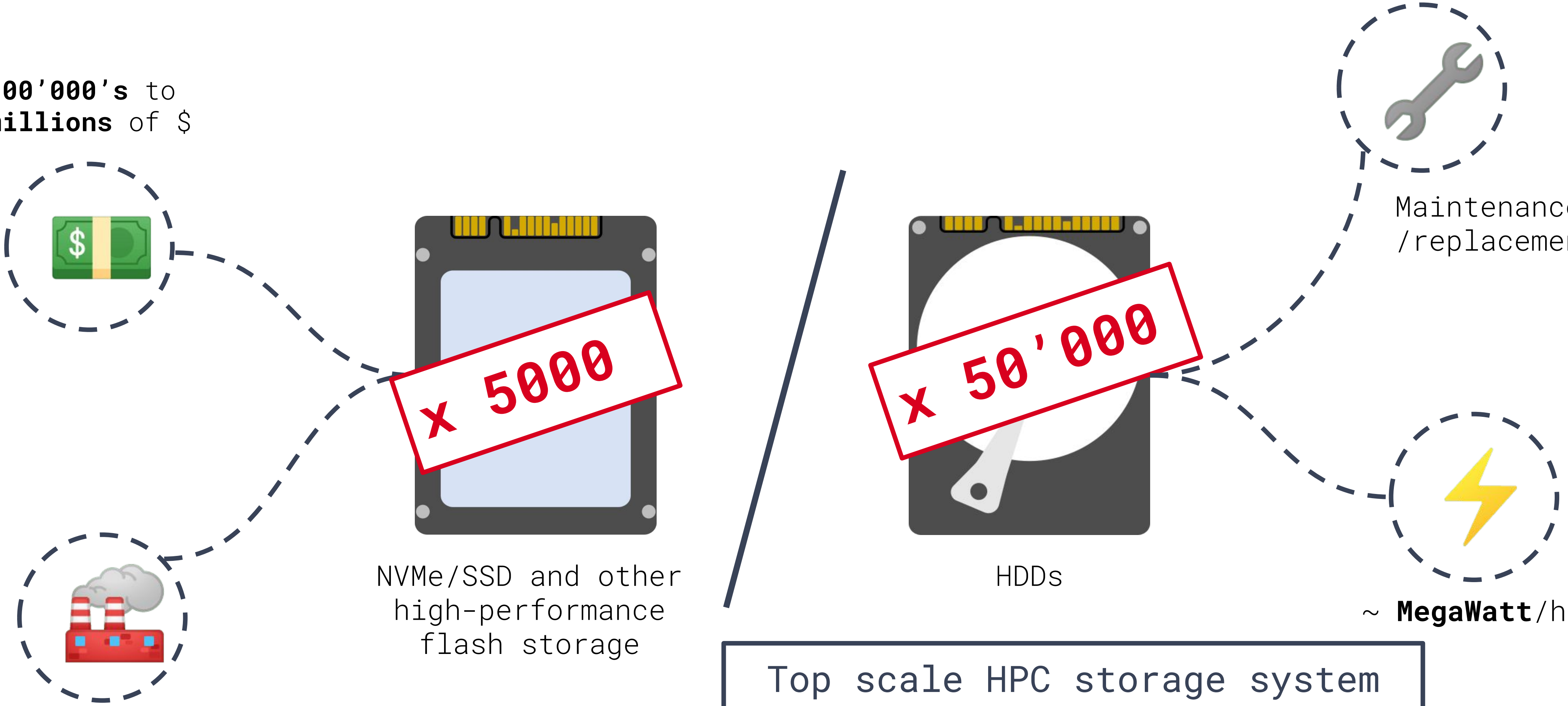
- **Migration** of HPC workflows/workloads **towards hybrid platforms**

Inria

Storage devices



Storage devices

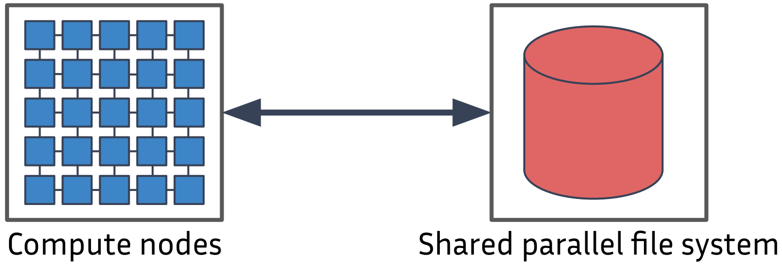


Top scale HPC storage system

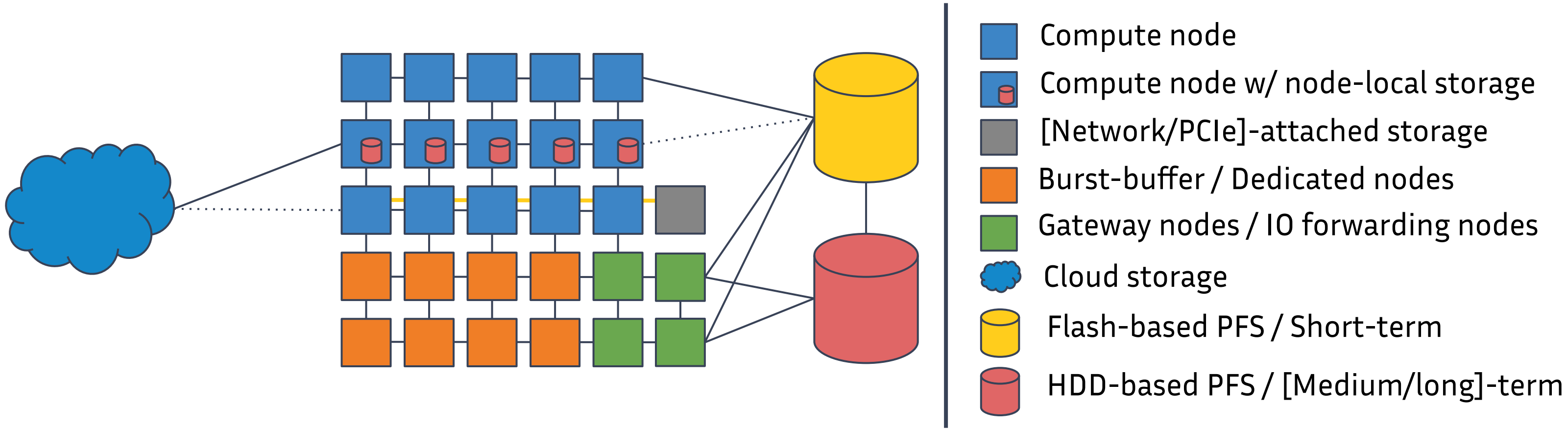
=

~ **5000** Flash storage drives
~ **50'000** HDDs

We went from traditional HPC storage systems...



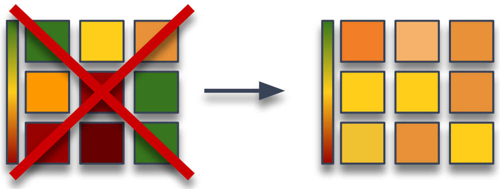
...to more complex and hybrid resources:



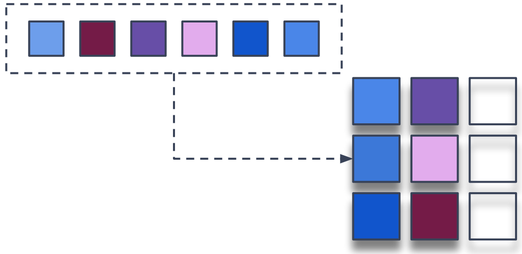
↗ **Complexity** and **underutilization** of resources

Problem statement

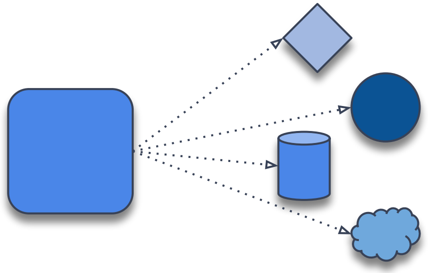
How can we leverage all available heterogeneous storage resources in order to maximize I/O efficiency?



Make **efficient** and **fair** use of all storage resources



Transparently allocate storage for users and applications



Deal with **heterogeneity of hardware resources**

Transpose **compute resource management knowledge** to storage resources

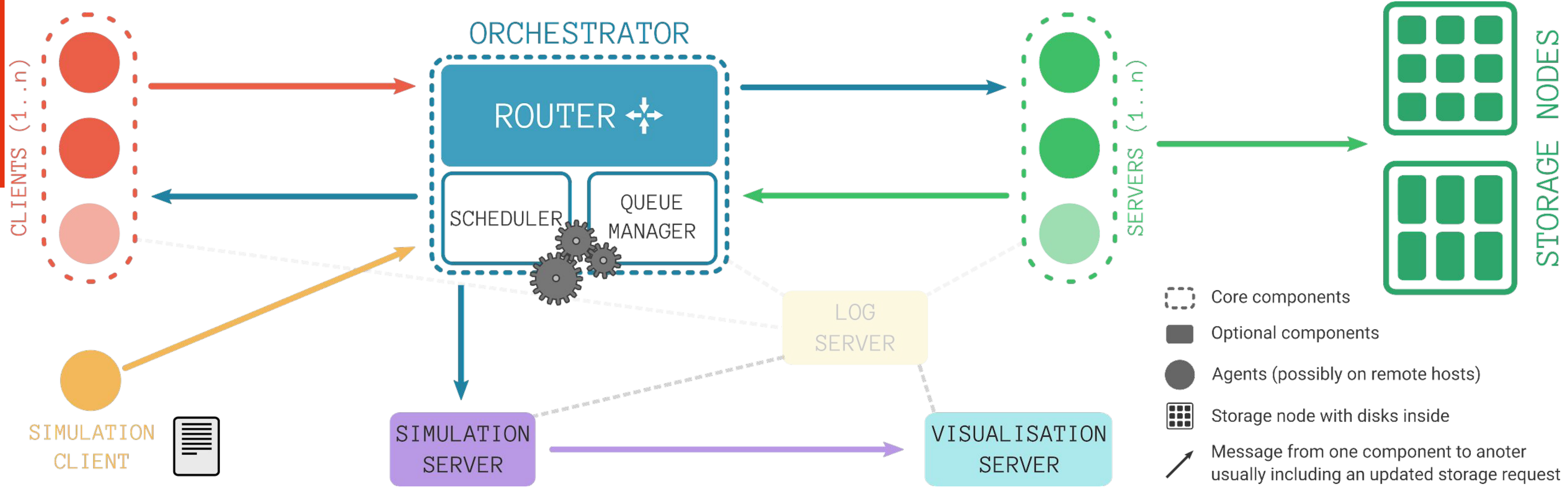
02

Our proposal: StorAlloc

StorAlloc

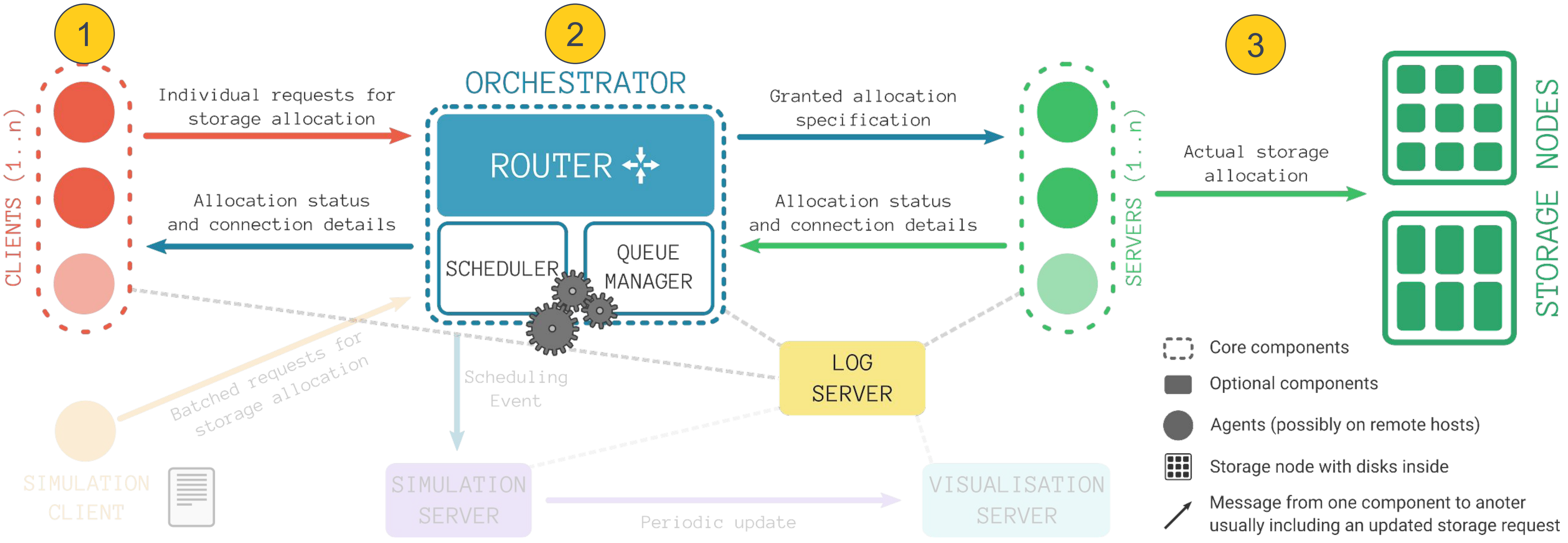
Simulation of storage-aware job scheduler

- → **Easy** implementation **of new scheduling algorithms**
- → Representation of **diverse storage technologies**
- → Detailed **simulation metrics**



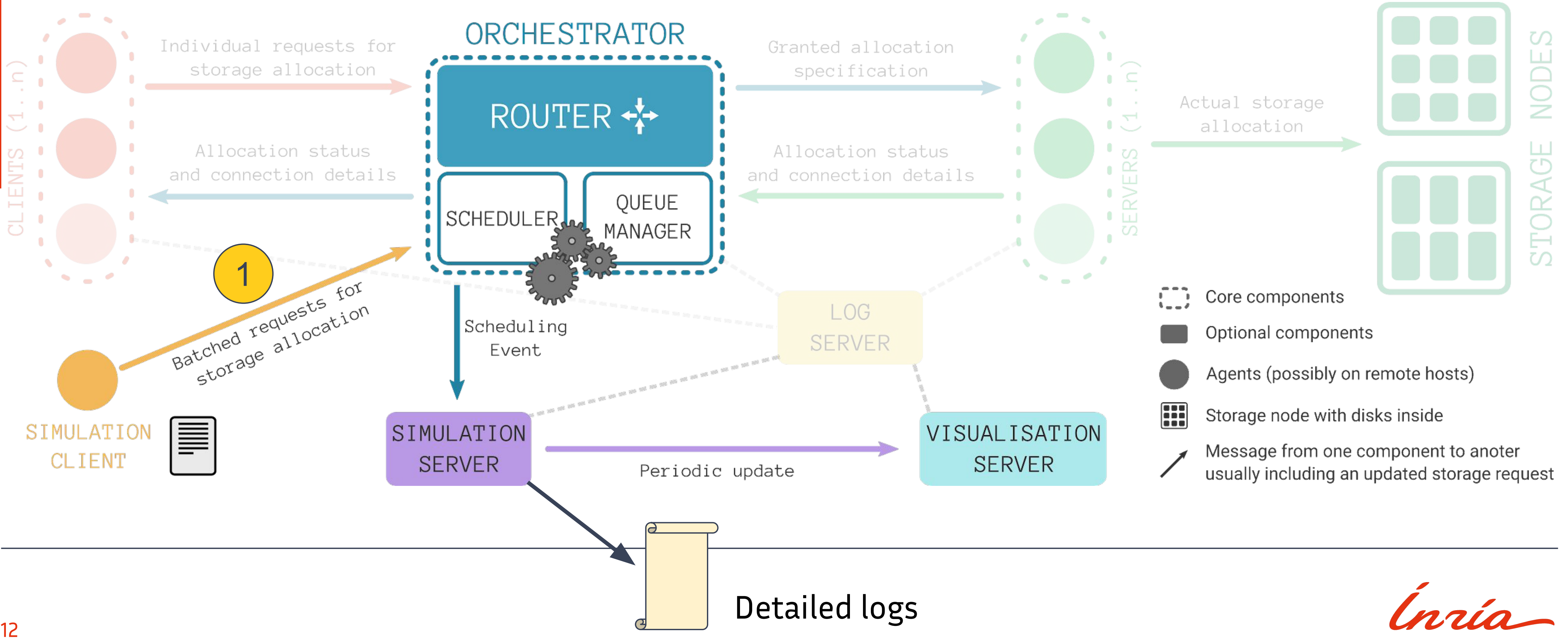
Core components

StorAlloc was initially designed as a middleware for partitioning and allocating network-attached storage



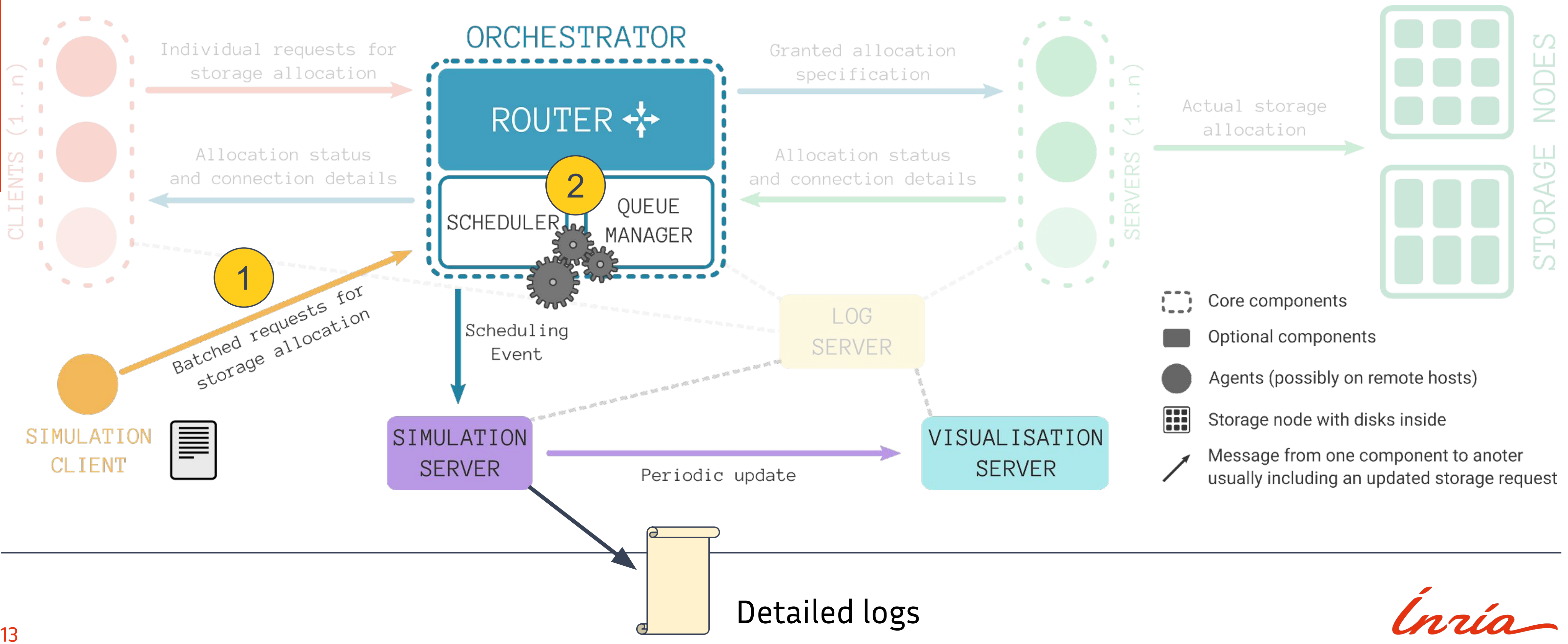
Simulation & visualization components

Simulation and visualisation capabilities are offered by optional components



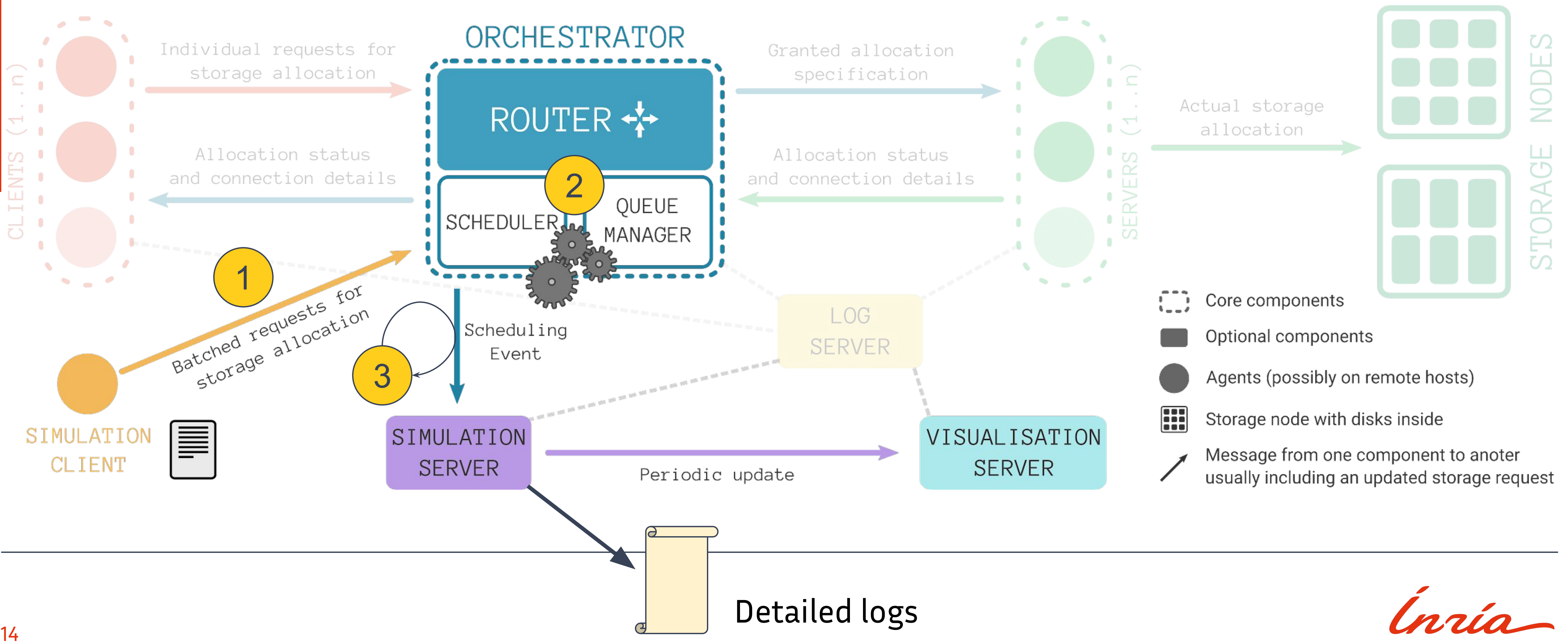
Simulation & visualization components

Simulation and visualisation capabilities are offered by optional components



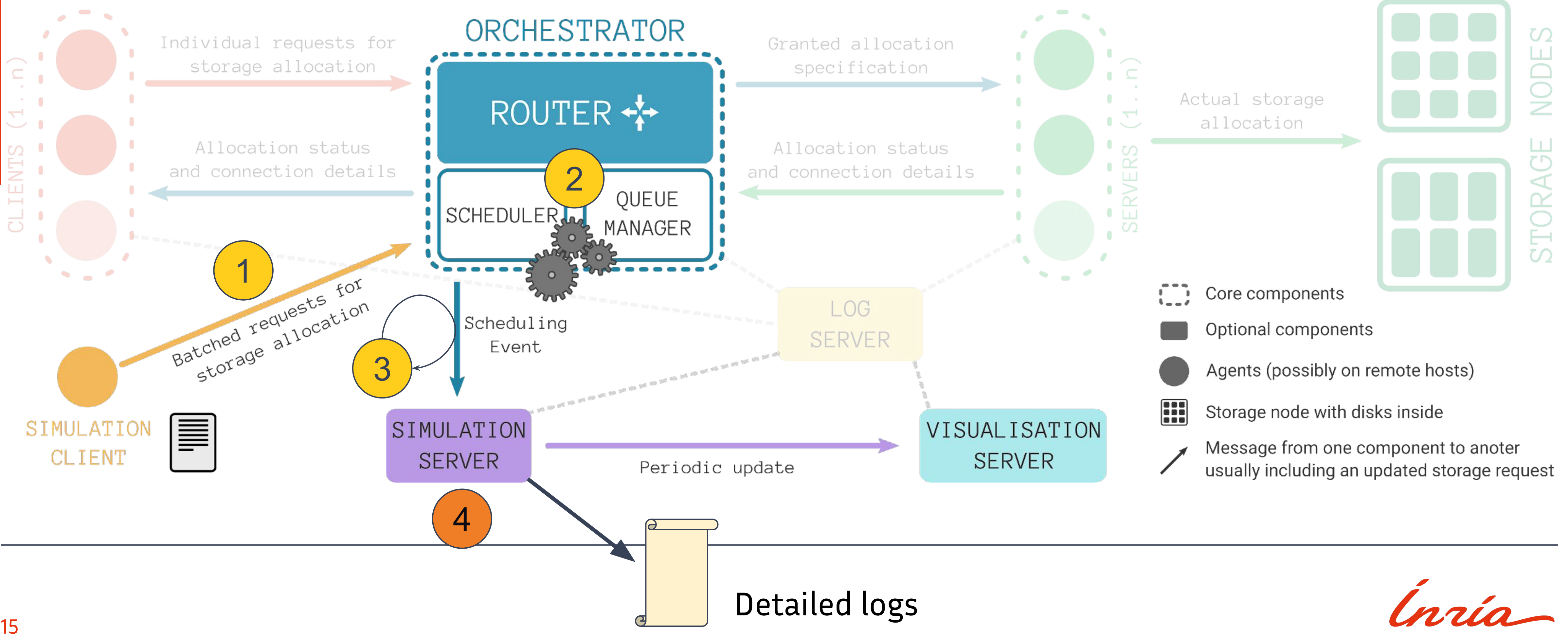
Simulation & visualization components

Simulation and visualisation capabilities are offered by optional components



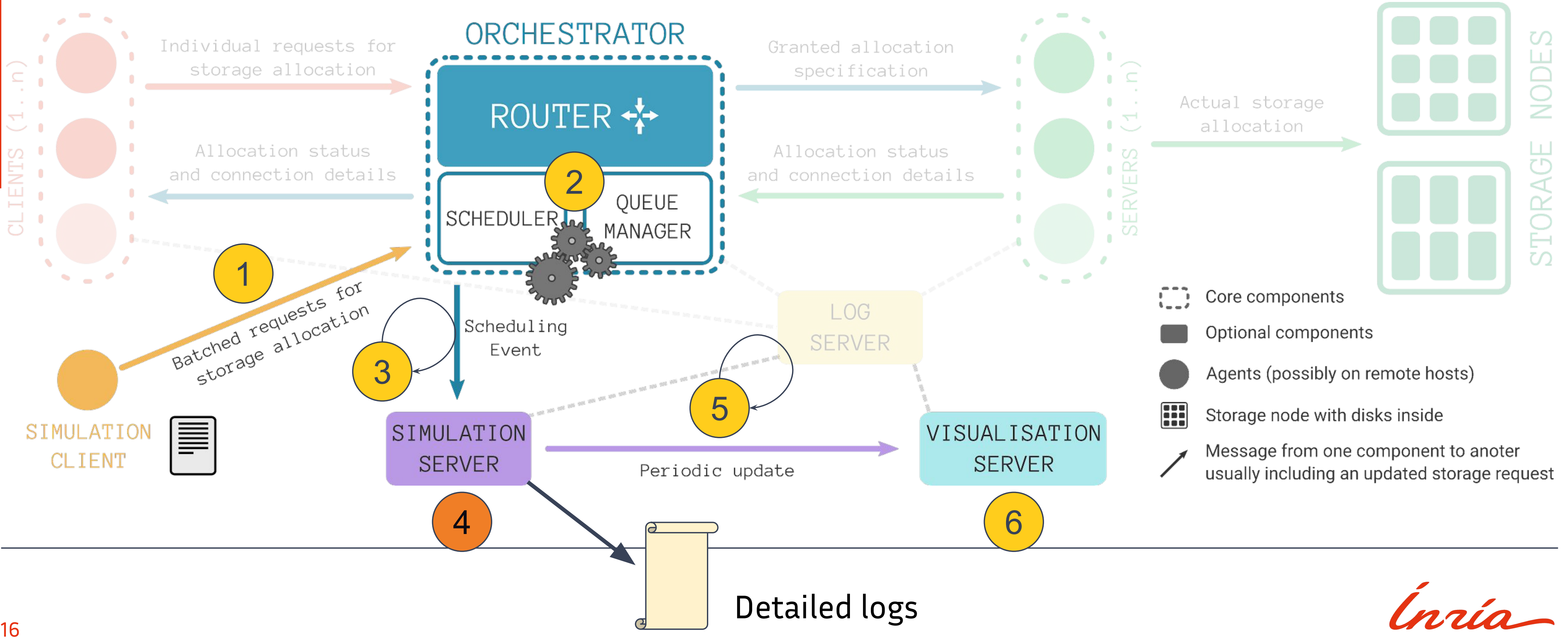
Simulation & visualization components

Simulation and visualisation capabilities are offered by optional components



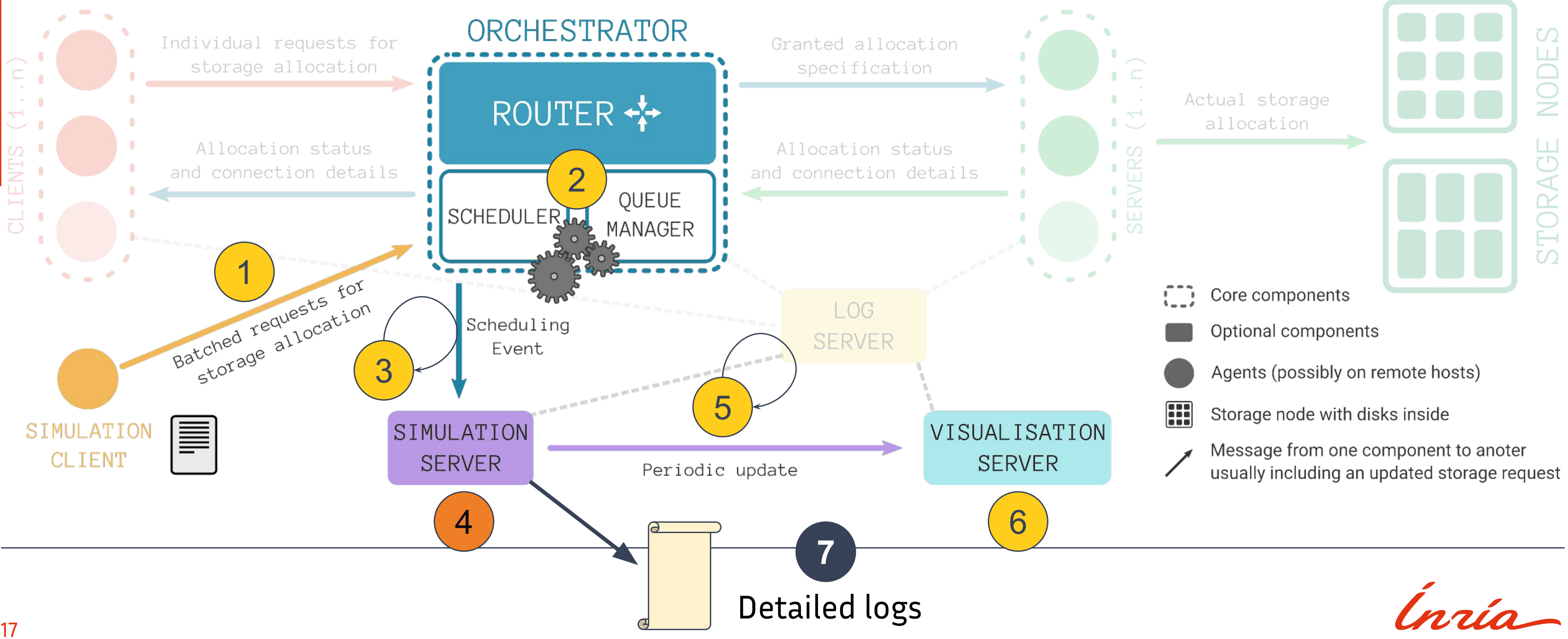
Simulation & visualization components

Simulation and visualisation capabilities are offered by optional components



Simulation & visualization components

Simulation and visualisation capabilities are offered by optional components

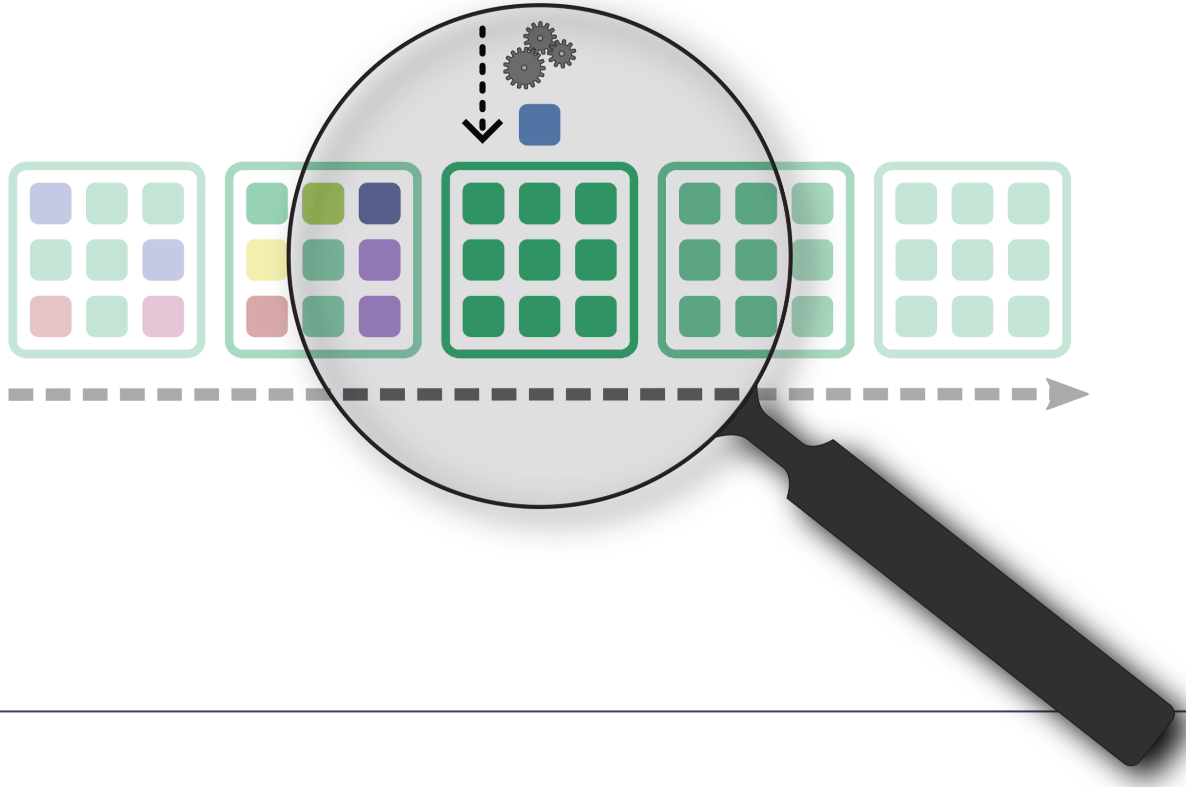


03

Experiments and results

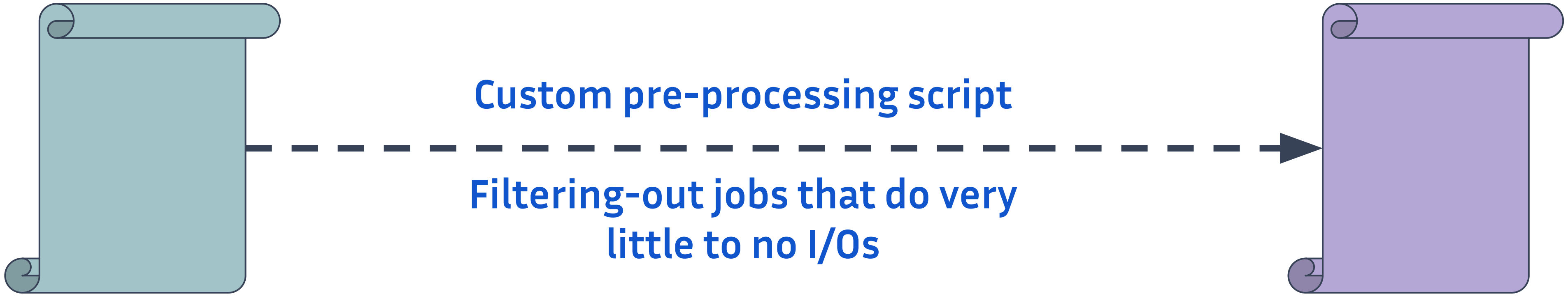
Experiments goal

Can we use StorAlloc simulations to get **useful** and **relevant insights** on the **storage allocations**?



Experimental protocol

We use a single dataset, extracted from processed Darshan traces, for all experiments



- Darshan I/O traces[1]
- From **Theta** (~12PFlops Cray XC40 supercomputer)
- 1 year
- ~ 624,000 jobs traced

- Job-level I/O data only
- 1 year
- ~ **24,000** "I/O-intensive" jobs

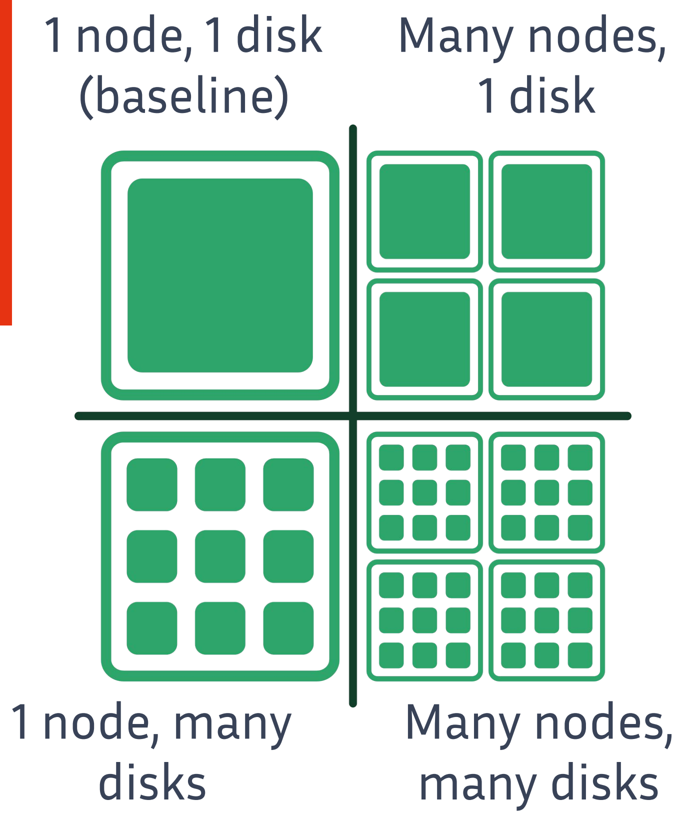
[1] HPC I/O characterization tool developed at ANL.

Argonne NATIONAL LABORATORY

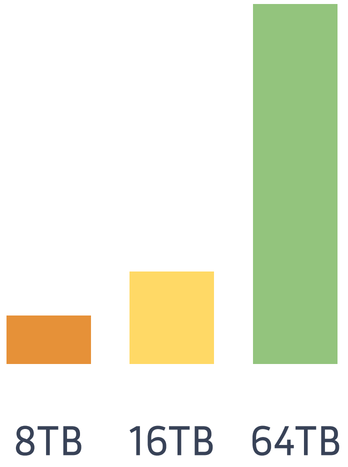
Experimental protocol

We run the allocation simulation several times on a laptop, for all combinations of parameters

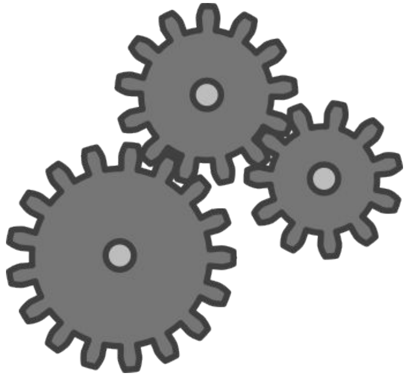
Storage Layout



Storage Capacity

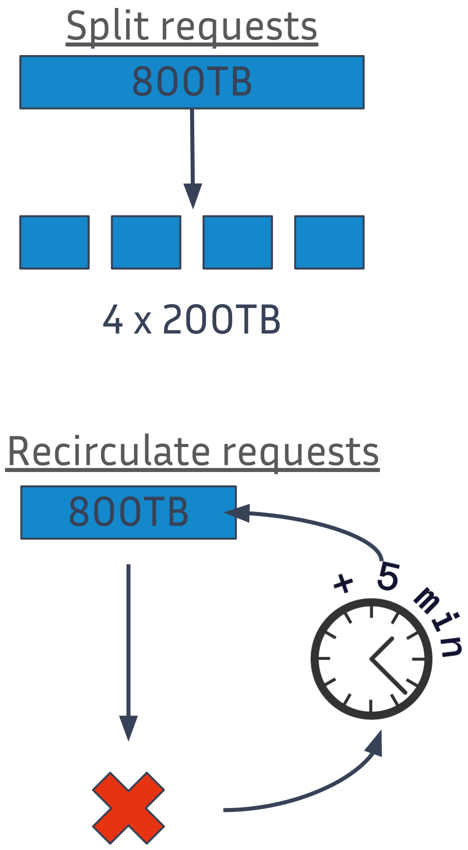


Algorithm

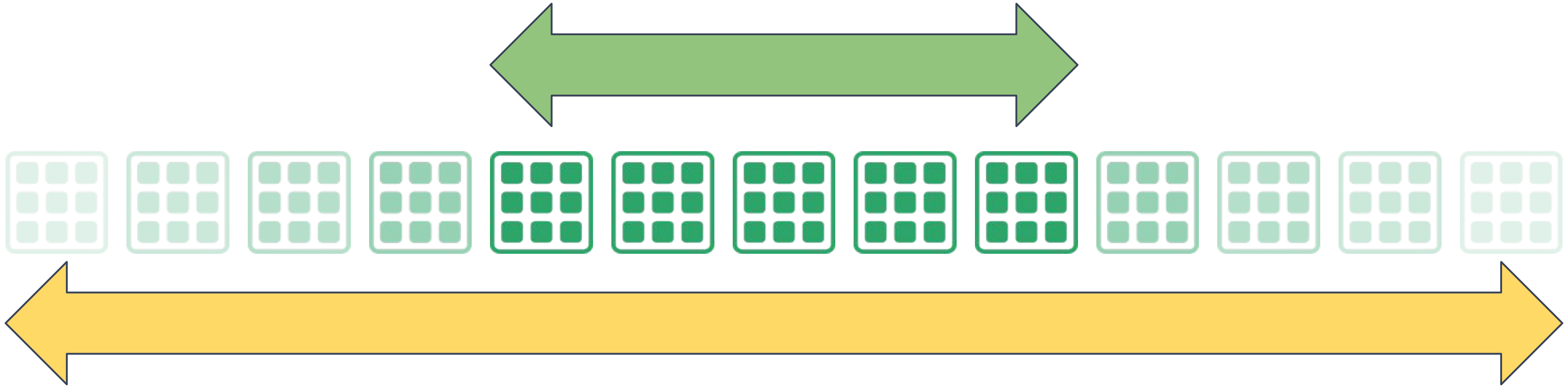


- Random
- Round-robin
- Worst-Fit
- Best-bandwidth

Strategy



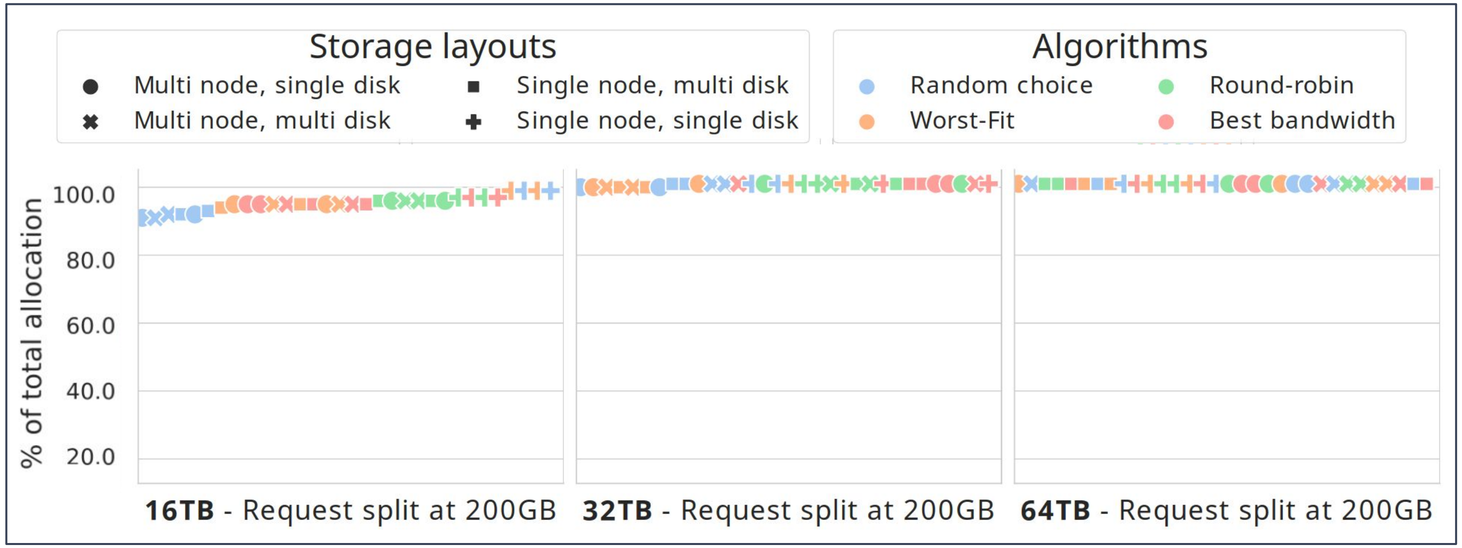
Given a dataset of jobs from Theta, can we determine a good fitting burst-buffer capacity for this platform?



Showcase analysis

What % of requested storage did we successfully allocate?

- Between ~80 and 90% for 16TB storage capacity
- ~100% for 32TB storage capacity with *Round-robin* and *Best bandwidth*
- ~100% for 64TB storage capacity



What is the maximum disk capacity utilisation at any time during simulation?

- ~100% for almost all disks for 16TB storage
- Above ~80% and below ~100% with *Round-robin* and *Best bandwidth* for 32TB storage
- Between ~40% and ~65%, depending on algorithm, for 64TB storage

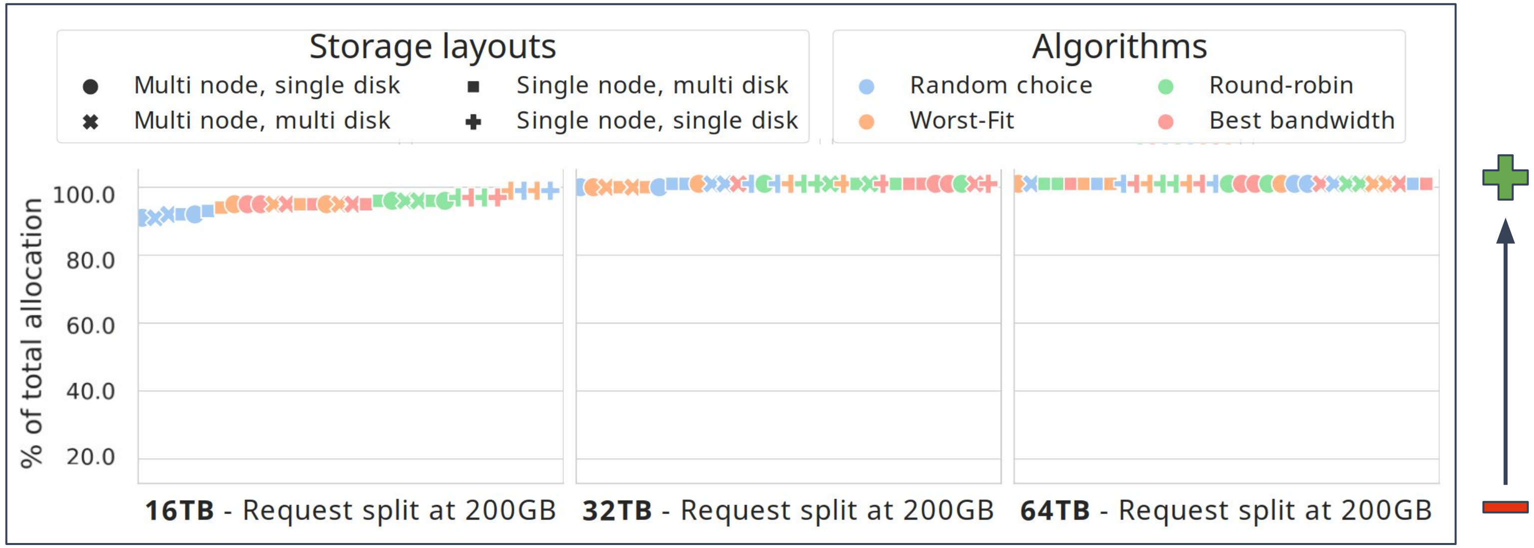
Disk full

Disk never full

Showcase analysis

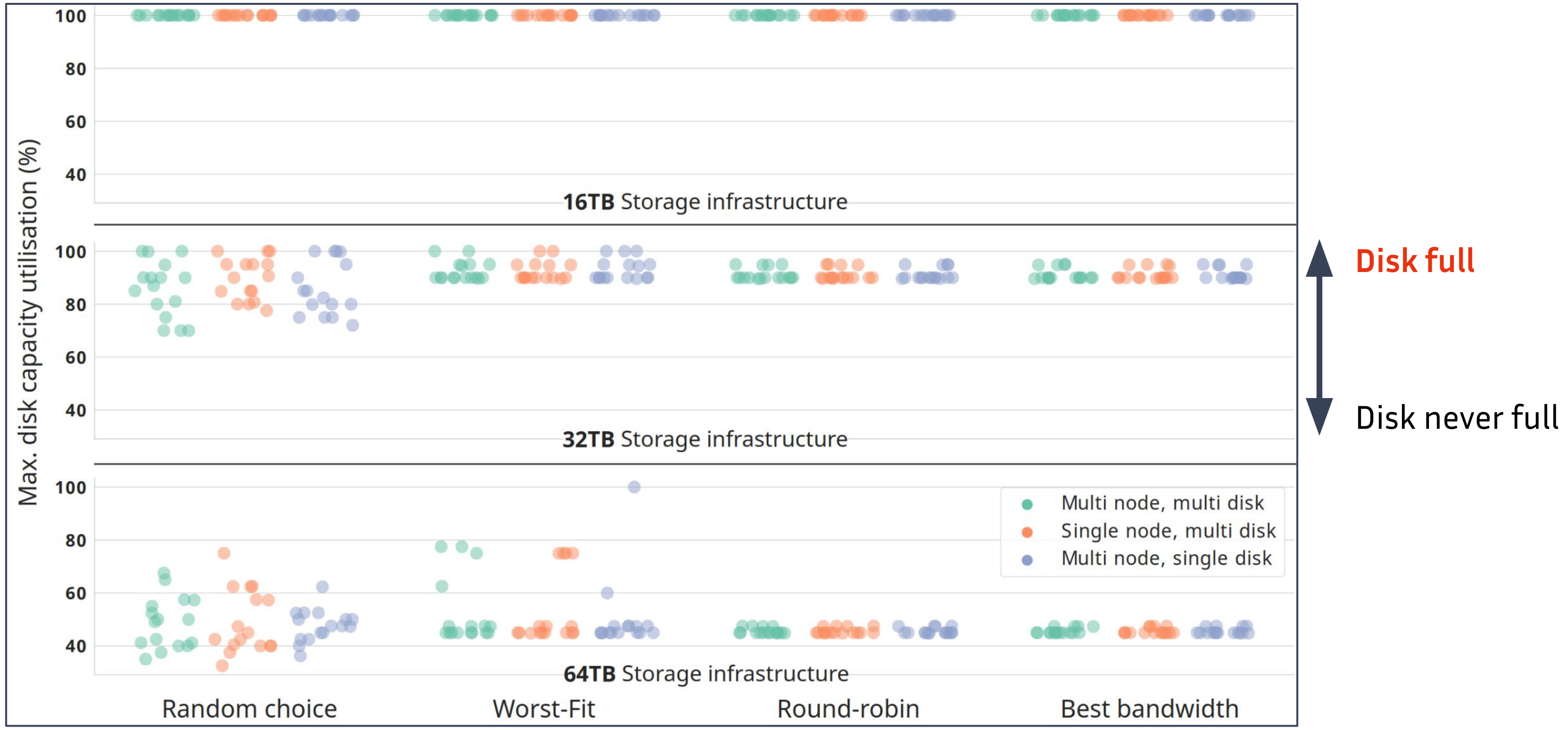
What % of requested storage did we successfully allocate?

- Between ~80 and 90% for 16TB storage capacity
- ~100% for 32TB storage capacity with *Round-robin* and *Best bandwidth*
- ~100% for 64TB storage capacity



What is the maximum disk capacity utilisation at any time during simulation?

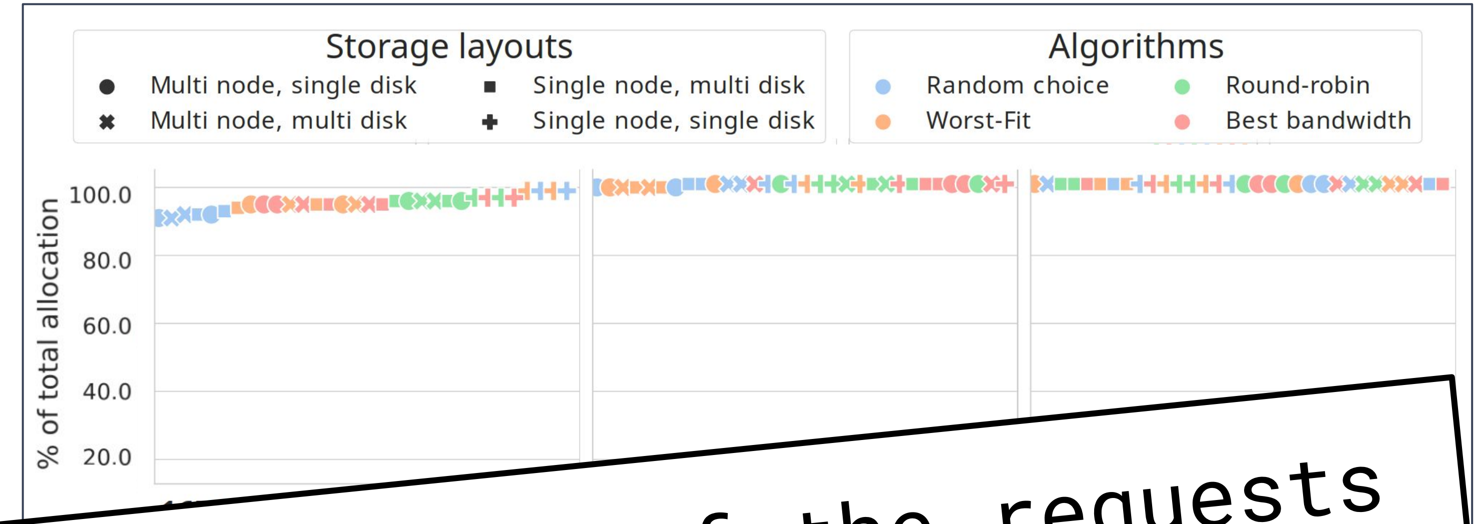
- ~100% for almost all disks for 16TB storage
- Above ~80% and below ~100% with *Round-robin* and *Best bandwidth* for 32TB storage
- Between ~40% and ~65%, depending on algorithm, for 64TB storage



Showcase analysis

What % of requested storage did we successfully allocate?

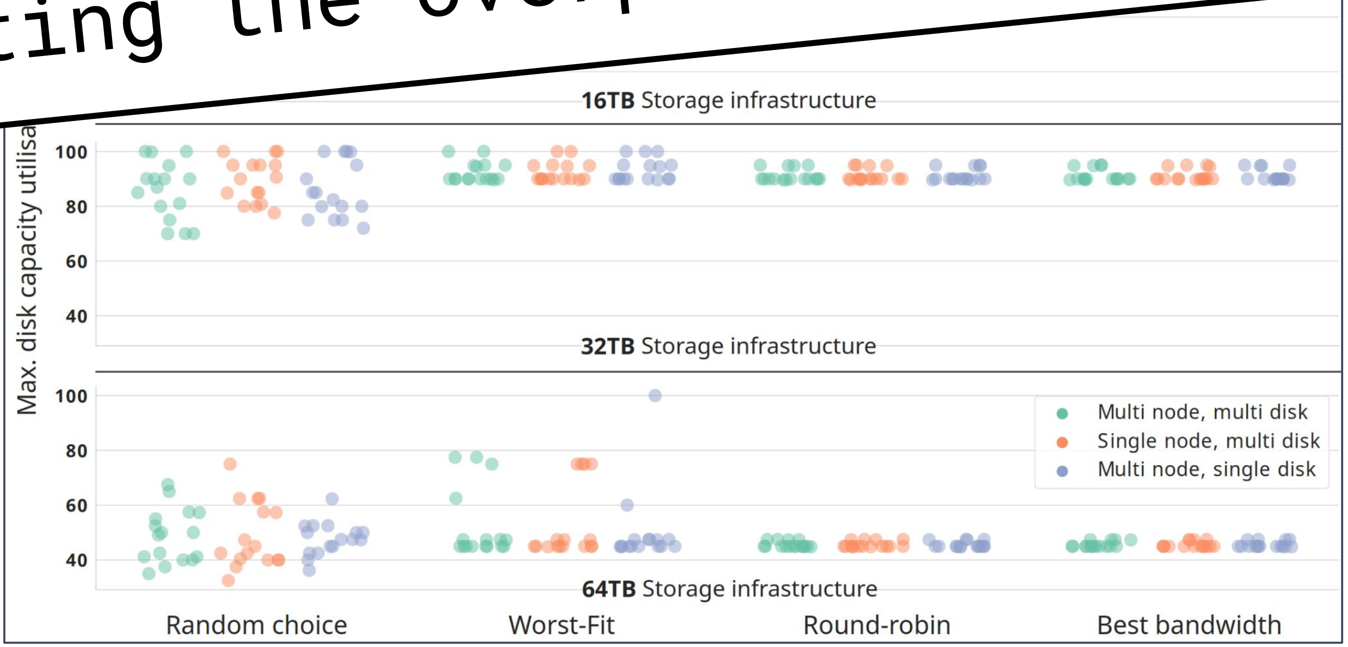
- Between ~80 and 90% for 16TB storage capacity
- ~100% for 32TB storage capacity with *Round-robin* and *Best bandwidth*
- ~100% for 64TB storage capacity



32TB → Can handle 100% of the requests while limiting the overprovisioning

What is the
at any time d

- ~100% for almost all disks for 16TB storage
- Above ~80% and below ~100% with *Round-robin* and *Best bandwidth* for 32TB storage
- Between ~40% and ~65%, depending on algorithm, for 64TB storage



Disk full

Disk never full

Related works 📚

- Simulators [1][2] for HPC platforms are often:
  - Rather **compute-centric**
  - **Not accounting for heterogeneity** of storage
  - Focused on data **movement**, not storage **provisioning**

- Some initiatives deal with heterogeneity and performance in storage, but **not as simulators or hybrid solutions** (DAOS, Rabbit, …)

- Some solutions focus on scheduling storage, but for a **single tier of storage** (eg. burst-buffer) [3][4][5]

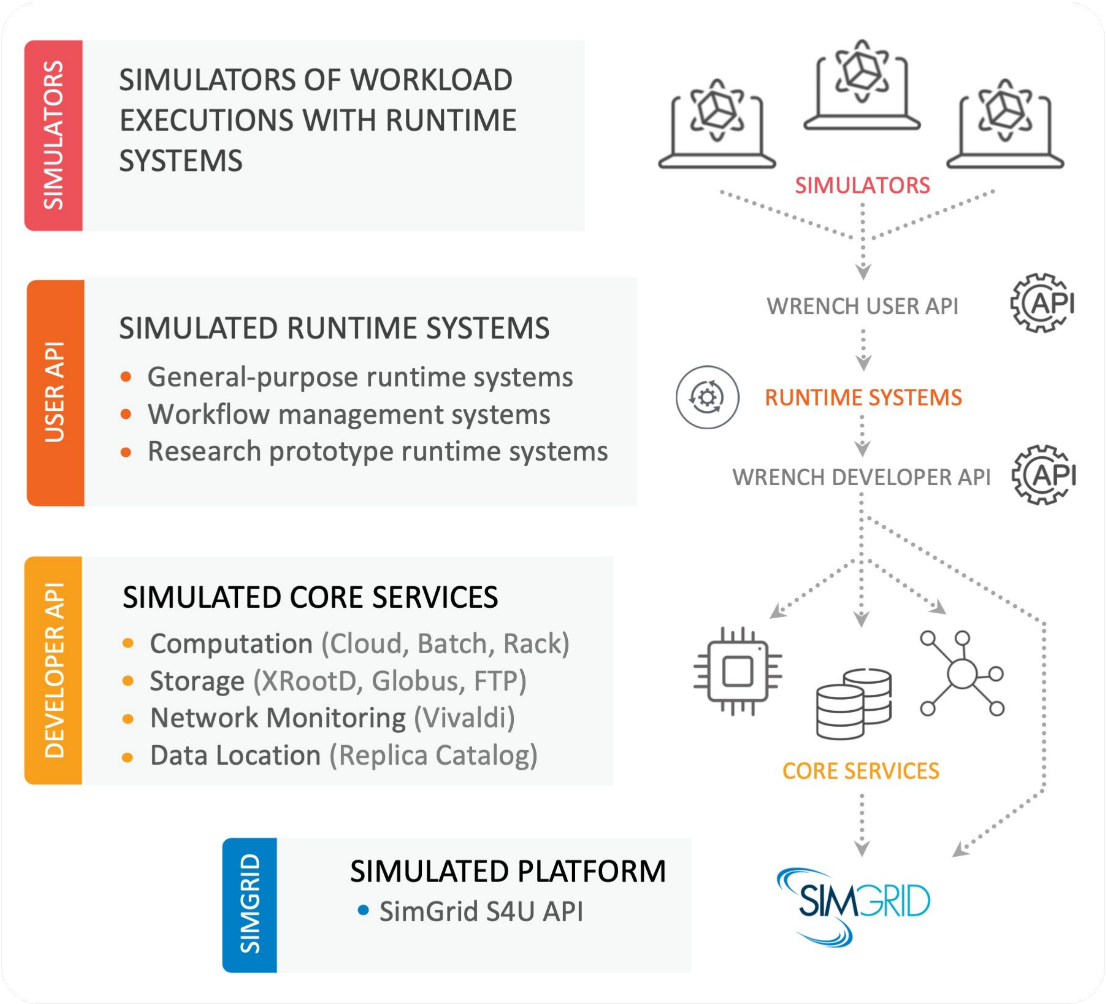
[1] *Versatile, Scalable, and Accurate Simulation of Distributed Applications and Platforms, H. Casanova et al.*
[2] *Developing Accurate and Scalable Simulators of Production Workflow Management Systems with WRENCH,H. Casanova et al.*
[3] *Dynamic Provisioning of Storage Resources: A Case Study with* ***Burst Buffers****, Tessier et al.*
[4] *Sizing and Partitioning Strategies for* ***Burst-Buffers*** *to Reduce IO Contention, Aupy et al.*
[5] *Automatic Dynamic Allocation of* ***Cloud Storage*** *for Scientific Applications, Al-Dhuraibi et al.*

03

What's next?

Porting StorAlloc to WRENCH…

WRENCH



https://wrench-project.org/wrench/2.1/

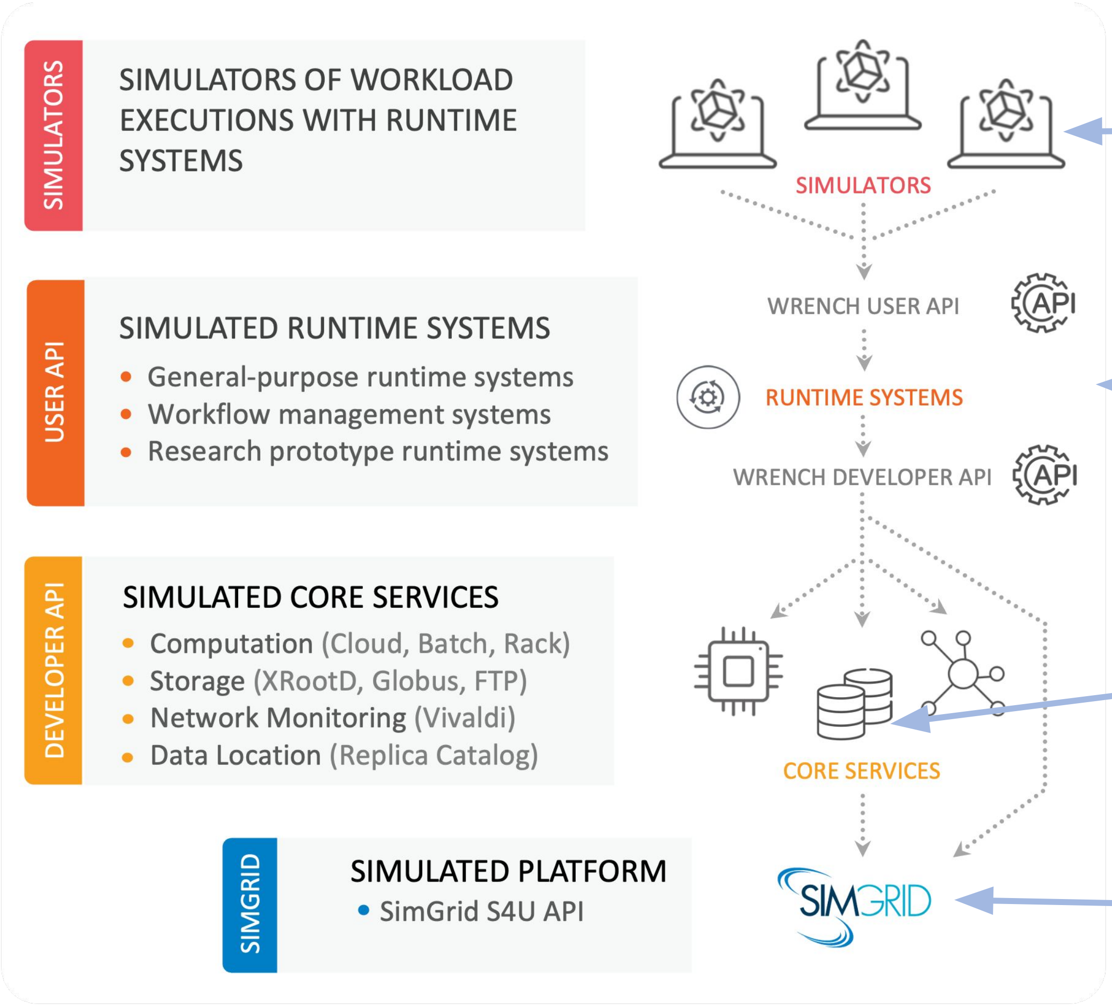
- **Linear system solver** instead of DES:
  - Feedback *during* simulation
  - Potential for better accuracy
- Has a **batch scheduler implementation** we can build upon

In collaboration with
Henri Casanova

University of Hawai'i Mānoa

Porting StorAlloc to WRENCH…

WRENCH



https://wrench-project.org/wrench/2.1/

Custom storage-aware simulator

Adding new storage-related logs

New Storage Service w/ user defined allocation function

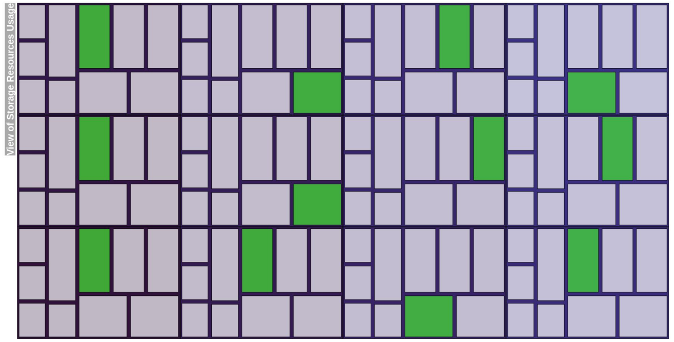
Custom variability and contention models thanks to Simgrid callbacks

Porting StorAlloc to WRENCH…

Observing the allocation algorithm behaviour throughout the simulation

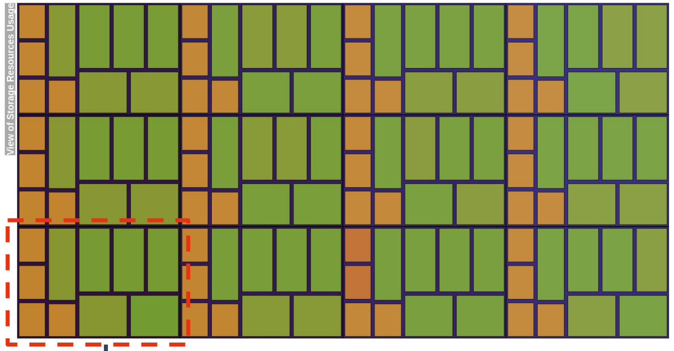
Treemap of basic **round-robin allocations** on heterogeneous storage **with file stripping**

Sparse allocations

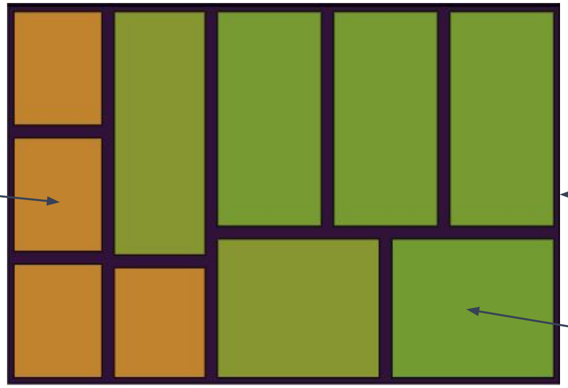


. . . .

Dense allocations



One storage node



04

Conclusions

Our contributions 📜

- Exploring methods for **dynamic allocation of heterogeneous storage resources**

- **StorAlloc: a simulation-based testbed** for scheduling algorithms and storage abstractions

- Preliminary validation: **useful insights on the allocation process,** using **data from actual job executions on Theta**

[1] Julien Monniot, François Tessier, Matthieu Robert, Gabriel Antoniu. StorAlloc: A Simulator for Job Scheduling on Heterogeneous Storage Resources. HeteroPar 2022, Aug 2022, Glasgow, United Kingdom.

Roadmap 🚧

- Ongoing integration with **state of the art simulation framework** (**WRENCH**)
- Study and development of **scheduling algorithms** for **storage resources**
- Integrate and test algorithms developed with StorAlloc in a resource manager such as **SLURM**

Thank you!

Takeway

POSTER 61
ROOM C-1-2-3

- StorAlloc supports research on dynamic allocation of heterogeneous storage resources
- It shows valuable insights on the allocation process using traces from actual HPC jobs
- Next steps: port StorAlloc features to WRENCH, and develop algorithms that we may test on a resource manager, such as SLURM

🐱 **Github repository:**
https://github.com/hephtaicie/storalloc

📧 **Contacts:**
{ julien.monniot,
francois.tessier,
gabriel.antoniu }@inria.fr

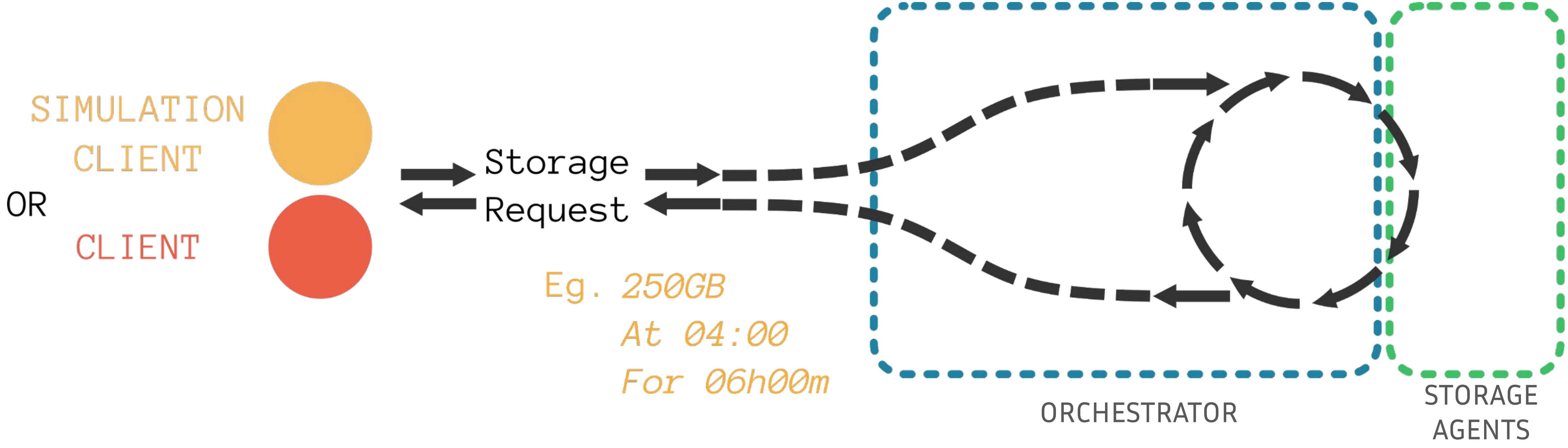
04

Extras

How to support user storage requirements?

Two kinds of messages:

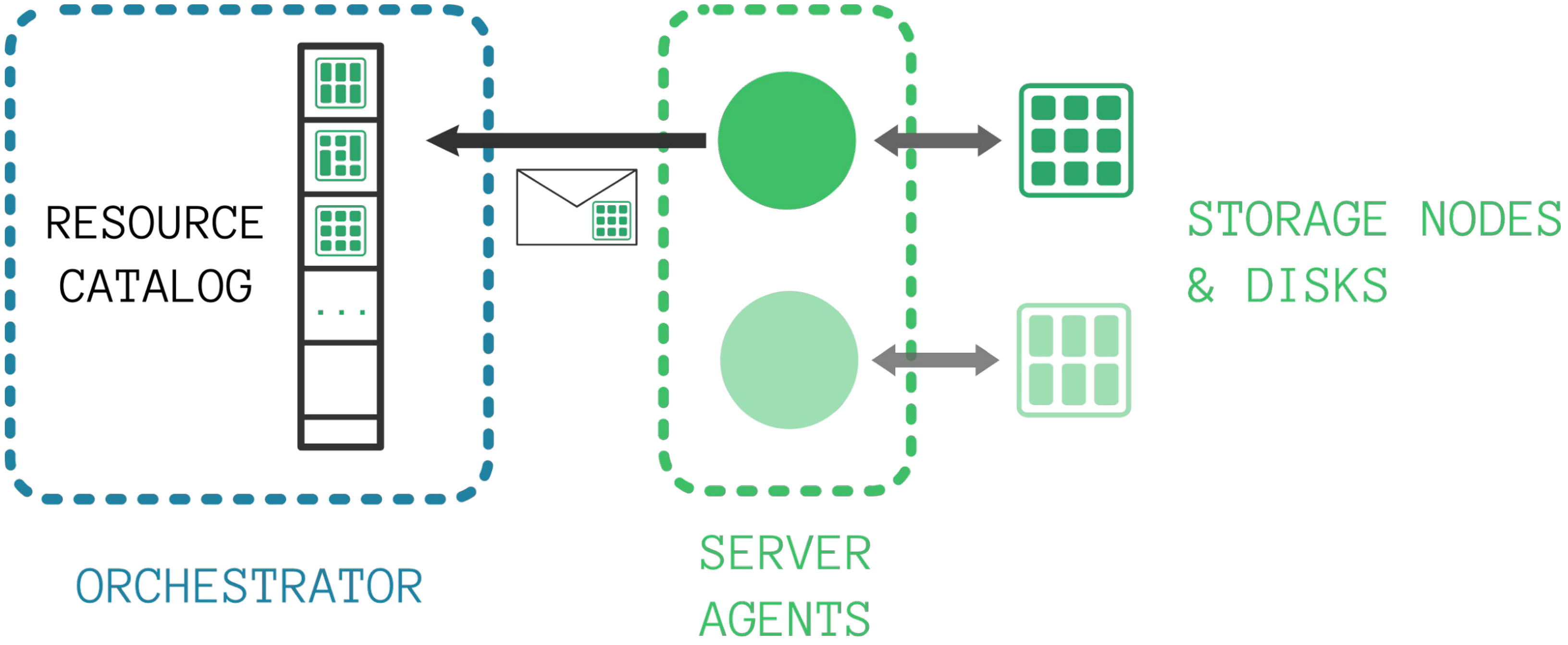
- Storage **request** → *Ask for some storage*
- Storage **registration** → *Declare that you can instrument some storage resources*



How to declare storage on the go?

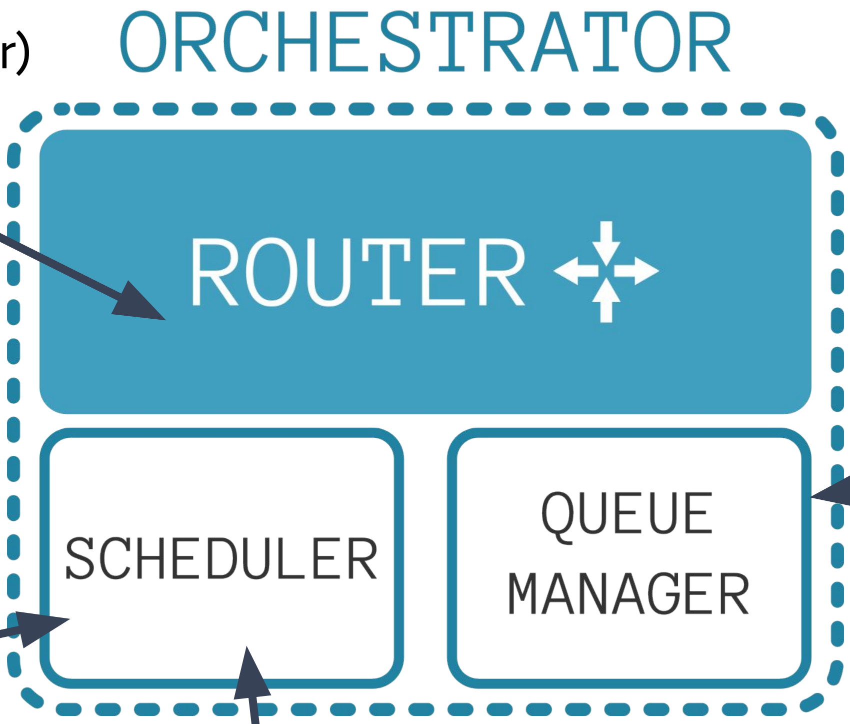
Two kinds of messages:

- Storage **request** → *Ask for some storage*
- Storage **registration** → *Declare that you can instrument some storage resources*



Detail: How to swap algorithms?

Message routing from and to any component
(inc. scheduler and queue manager)



Triggers deallocations

Schedules storage requests.
Algorithm chosen from config file upon startup

Keeps tracks of available resources

Detail: How to abstract heterogeneous storage?

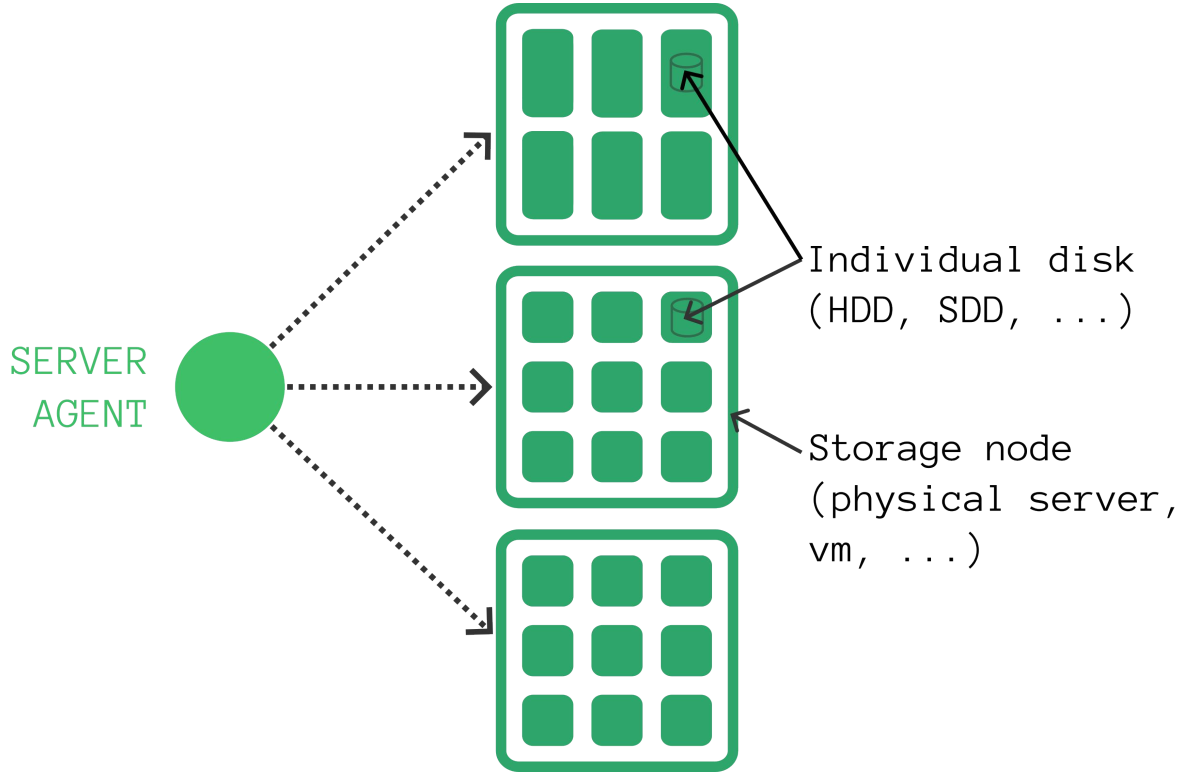
Available fields

Nodes

- **Network bw**
- List of **disks**
- [Hostname]
- [IP]
- ...

Disks

- **Capacity**
- Read / Write **bw**
- Serial
- Hardware type
- Model / maker
- ...



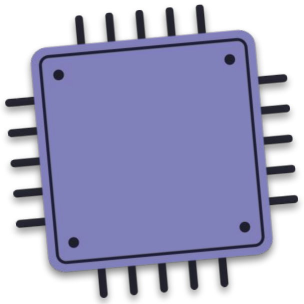
Research direction

The case for dynamic allocation of heterogeneous storage resources:

- **Reconfigure** storage on the fly (*hardware level*)
    - **Easier integration** and use of new storage technologies

- **Holistic view** of contention areas

- **Single interface** for giving access to raw storage
    - User/application/middleware may get full control of allocated resources



Transpose **compute resource management knowledge** to storage resources